

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-27666

(43) 公開日 平成11年(1999) 1月29日

(51) Int.Cl.⁸

識別記号

F I

H 0 4 N 7/24

H 0 3 M 7/30

H 0 4 N 1/41

H 0 4 N 7/13

H 0 3 M 7/30

H 0 4 N 1/41

Z

B

B

審査請求 未請求 請求項の数 3 O L (全 8 頁)

(21) 出願番号

特願平9-175443

(22) 出願日

平成9年(1997) 7月1日

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 下田 晴朗

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 野間 睦明

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 河野 昭宏

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(74) 代理人 弁理士 滝本 智之 (外1名)

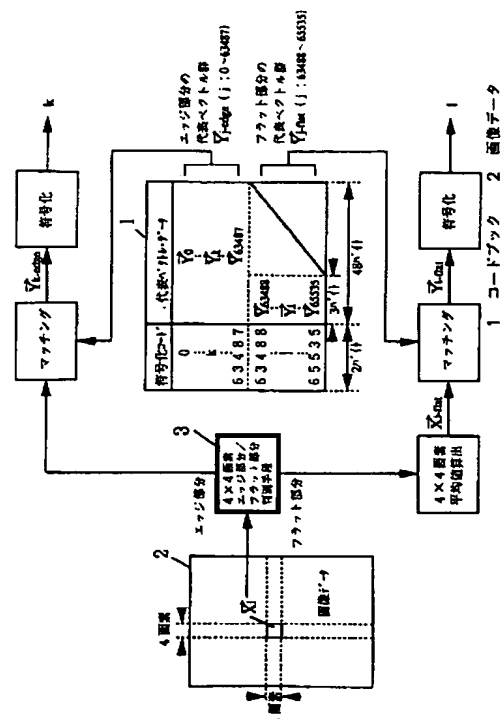
最終頁に続く

(54) 【発明の名称】 画像符号化方法

(57) 【要約】

【課題】 圧縮画像の画質を落とさず画像全体の冗長度を削減できる画像符号化方法を提供することを目的とする。

【解決手段】 画像をある画素数のブロックに分割しベクトル量子化により圧縮する場合に、エッジ部では画質を落とさずフラット部では情報量をより小さくするため、 k 個の色成分を持つ画素からなるデジタル画像データを $m \times n$ 画素ブロックの画像ベクトルに分割してベクトル量子化により圧縮する際に、各画素ブロックの画像ベクトルがエッジ部分なのかフラット部分なのかを判別し、エッジ部分に適合するように定めた符号化方式とフラット部分に適合するように定めた符号化方式との、いずれかの符号化方式に基づいて符号化する。



【特許請求の範囲】

【請求項 1】 デジタル画像データをベクトル量子化により圧縮する画像符号化方法であって、 k 個の色成分を持つ画素からなるデジタル画像データを $m \times n$ 画素ブロックの画像ベクトルに分割してベクトル量子化により圧縮する際に、各画素ブロックの画像ベクトルがエッジ部分なのかフラット部分なのかを判別し、エッジ部分に適合するように定めた符号化方式とフラット部分に適合するように定めた符号化方式との、いずれかの符号化方式に基づいて符号化することを特徴とする画像符号化方法。

【請求項 2】 エッジ部分について、 $m \times n$ 画素ブロックに対する $m \times n \times k$ 次元のベクトル量子化を行う符号化方式が定められ、フラット部分について、 $m \times n$ 画素ブロックの平均値画素に対する k 次元のベクトル量子化を行う符号化方式が定められていることを特徴とする請求項 1 記載の画像符号化方法。

【請求項 3】 $p \times q$ 個の $m \times n$ 画素ブロックの全てがフラット部分であるとき、これらの $m \times n$ 画素ブロックの全てを一つのブロックとみなしフラット部分について定められた符号化方式に基づいて符号化することを特徴とする請求項 1 記載の画像符号化方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、デジタル画像データの画像符号化方法に関するものである。

【0002】

【従来の技術】 従来から画像は、伝送・保存時等の情報量を減らすために、デジタル・データに置き換えられ符号化・圧縮されている。特に最近では、取り扱われる画像のカラー化・精細化により画像に含まれる情報量が大幅に増加しており、これに対応して各種の画像符号化方法が標準化されている。しかし、さらなる伝送時間の短縮や蓄積画像数の向上のために、より高圧縮かつ高画質な符号化方法に対する要求も高まりつつある。その 1 つとして、ベクトル量子化を用いた画像符号化方法が注目されている。

【0003】 以下に、従来の画像符号化方法について説明する。図 3 は、従来の画像符号化方法のプロセス説明図である。まず、図 3 の左側に示すように、入力された画像データをある画素数（例えば 2×2 画素）からなるブロック毎に分割し、それぞれのブロックに含まれている画像情報（色、輝度、色差など）を 1 つの画像ベクトルとみなす。これにより画像データは、 r 個の画像ベクトル (X_1, \dots, X_r) で表現できる。

【0004】 次に、図 3 の上側に示すように、これらの画像ベクトル (X_1, \dots, X_r) を置き換えるため、 s 個の代表ベクトル (Y_1, \dots, Y_s) (但し $s < r$) からなるコードブックを作成する。そして、それぞれの画像ベクトルをコードブック内の各代表ベクトルと比較

し、最も類似した代表ベクトルを選択する。つまり、ある画像ベクトル X_i ($i: 1 \dots r$) に対して、最もノルム距離が近い代表ベクトル Y_j ($j: 1 \dots s$) を選択する。ここで、ノルム距離とは、画像ベクトルと代表ベクトルとのそれぞれの成分について両者の差の 2 乗を求め、それらを足し合わせたものの平方根を求めた値のことである。こうして、画像ベクトル X_i で表現されていた該当ブロックは、最もノルム距離が近い代表ベクトル Y_j に置き換えられて表現される。

【0005】 以上の様な方法で画像を符号化すれば、画像データ (X_1, \dots, X_r) は、代表ベクトル (Y_1, \dots, Y_s) (但し $s < r$) からなるコードブックと、各ブロックがどの代表ベクトルで表現されるかという情報とからなるデータに置き換えられ、全体のデータ量を圧縮することができる。ここで、一般的なベクトル量子化では、符号化の単位となるブロック内の画素数は、例えば $m \times n$ 画素という様に、あらかじめ 1 つに決められている。1 画素が k 次元の色空間で表わされているとすれば、この場合は $m \times n \times k$ 次元のベクトル量子化を行うことになる。ベクトル量子化の次元数や符号化の単位となるブロック内の画素数は、画像データの各ブロックがどのような性質の画像であっても同じものが用いられる。

【0006】

【発明が解決しようとする課題】 しかしながら、従来の画像符号化方法において、符号化の単位となるブロックの画素数を小さく取ったのでは、画質は確保できるが圧縮率が上がらない。逆に、ブロックの画素数を大きく取ると圧縮率は上がるが、画質が極端に落ちるという欠点があった。これは、各ブロックの画像の性質によって落とすべき冗長度が異なるにもかかわらず、同じ仕様のベクトル量子化により各ブロックを取り扱って冗長度を同一に落としていたためである。

【0007】 ここで、画像があまり変化しない部分（フラット部分）では、そこに含まれる情報量をかなり落としても画像全体の画質には大きく影響しない（つまり冗長度が大きい）が、画像が細かく変化する部分（エッジ部分）では、情報量を落とす過ぎると画質が損なわれてしまう（つまり冗長度が小さい）。このため、一つの方式のみで構成される一般的なベクトル量子化で圧縮画像の画質を確保しようとするれば、あまり情報量を落とすことができない（つまり冗長度が小さい）エッジ部分に合わせて仕様を決定しなければならなかった。また、従来の画像符号化方法では、符号化の際にすべての次元の情報をコードブック内の各代表ベクトルと比較しなければならないため、符号化の単位となるブロックの画素数を大きく取れば取るほど符号化に時間がかかるという欠点を有していた。

【0008】 そこで本発明は、圧縮画像の画質を大きく落とすことなく画像全体の冗長度を削減することがで

き、かつ符号化時間を短縮し圧縮率を上げることができる、ベクトル量子化による画像符号化方法を提供することを目的とする。

【0009】

【課題を解決するための手段】本発明の第1の構成では、 k (k は自然数)個の色成分を持つ画素からなるデジタル画像データを $m \times n$ (m, n は自然数)画素ブロックの画像ベクトルに分割してベクトル量子化により圧縮する際に、各画素ブロックの画像ベクトルがエッジ部分なのかフラット部分なのかを判別し、エッジ部分に適合するように定めた符号化方式とフラット部分に適合するように定めた符号化方式との、いずれかの符号化方式に基づいて符号化する。この構成により、それぞれのブロックの画像の性質に合った適正な冗長削減度を持つ符号化方法を選択でき、圧縮画像を大きく落とすことなく画像全体の冗長度を削減できる。

【0010】また、本発明の第2の構成では、エッジ部分について、 $m \times n$ 画素ブロックに対する $m \times n \times k$ 次元のベクトル量子化を行う符号化方式が定められ、フラット部分について、 $m \times n$ 画素ブロックの平均値画素に対する k 次元のベクトル量子化を行う符号化方式が定められている。この構成により、ベクトル量子化の符号化時間を従来よりも短縮できる。さらに、本発明の第3の構成では、 $p \times q$ (p, q は自然数)個の $m \times n$ 画素ブロックの全てがフラット部分であるとき、これらの $m \times n$ 画素ブロックの全てを一つのブロックとみなしフラット部分について定められた符号化方式に基づいて符号化する。これにより、画像データの符号化圧縮率をさらに向上できる。

【0011】

【発明の実施の形態】請求項1記載の発明は、 k (k は自然数)個の色成分を持つ画素からなるデジタル画像データを $m \times n$ (m, n は自然数)画素ブロックの画像ベクトルに分割してベクトル量子化により圧縮する際に、各画素ブロックの画像ベクトルがエッジ部分なのかフラット部分なのかを判別し、エッジ部分に適合するように定めた符号化方式とフラット部分に適合するように定めた符号化方式との、いずれかの符号化方式に基づいて符号化する。この構成により、それぞれのブロックの画像の性質に合った適正な冗長削減度を持つ符号化方法を選択でき、圧縮画像の画質を大きく落とすことなく画像全体の冗長度を削減できる。

【0012】請求項2記載の発明は、エッジ部分について、 $m \times n$ 画素ブロックに対する $m \times n \times k$ 次元のベクトル量子化を行う符号化方式が定められ、フラット部分について、 $m \times n$ 画素ブロックの平均値画素に対する k 次元のベクトル量子化を行う符号化方式が定められている。この構成により、ベクトル量子化の符号化時間を従来よりも短縮することができる。

【0013】請求項3記載の発明は、 $p \times q$ (p, q は

自然数)個の $m \times n$ 画素ブロックの全てがフラット部分であるとき、これらの $m \times n$ 画素ブロックの全てを一つのブロックとみなしフラット部分について定められた符号化方式に基づいて符号化する。これにより、画像データの符号化圧縮率をさらに向上できる。

【0014】以下、本発明の実施の形態について、図面を参照しながら説明する。まず、図1を用いて、本発明の第1の実施の形態における画像符号化方法のプロセス説明図である。さて、本形態では、 $k = 3$ 、 $m = n = 4$ 、 $p = q = 2$ の場合について考える。すなわち、各画素は3成分の色空間で表現されており、符号化の単位となるブロックの画素数を 4×4 画素とし、 4×4 画素の各ブロック毎にエッジ部分なのかフラット部分なのかを判定すると共に、さらに 8×8 画素毎にフラット部分かどうかを判定する。それぞれの画素の色成分は、1バイト(256階調)で表現されているものとする。

【0015】また、本実施の形態のベクトル量子化のコードブック1は1つとし、その中に含まれる代表ベクトルの合計数は65536個とする。このため、符号化データは、1画像ベクトル当たり2バイトとなる。原画像データのすべての 4×4 画素ブロック画像ベクトル(48バイト)は、それぞれコードブック内のどれか1つの代表ベクトルに置き換えられ、その代表ベクトルが持つ0番から65535番までのどれか1つの番号、すなわち符号化コード(2バイト)で符号化される。

【0016】図1に示すように、コードブック1には、 4×4 画素エッジ部分の代表ベクトル群・ 4×4 画素フラット部分の代表ベクトル群をあらかじめ登録しておく。それぞれの代表ベクトル群の数および符号は、例えば 4×4 画素エッジ部分：63488個(0番～63487番)・ 4×4 画素フラット部分：2048個(63488番～65535番)とする。このようにすれば、符号を見るだけで、それが 4×4 画素のエッジ部分を表わすものか、 4×4 画素フラット部分を表わすものかわかる。

【0017】符号化の際には、図1の左側に示すように、まず画像データ2を 4×4 画素のブロック X_i

($i : 1 \dots r$)毎に順に見ていき、それぞれのブロックでエッジ部分なのかフラット部分なのかを、 4×4 画素エッジ部分/フラット部分判別手段3により判別する。 4×4 画素エッジ部分/フラット部分判別手段3は、例えば、ブロック内の全画素の輝度成分に注目し、その中から輝度の最大値と最小値を取りだしてその差分値を求め、その差分値をあらかじめ設定された閾値と比較して、差分値が閾値よりも小さければフラット部分と判定し、大きければエッジ部分と判定する。

【0018】エッジ部分と判定されたブロックについては、当該ブロックの全画素全成分、すなわち本実施の形態においては(4×4 画素) $\times 3$ 成分を1つの48次元

の画像ベクトル X_{i-edge} ($=X_i$) としてベクトル量子化を行う。これにより、画質の低下を防止できる。具体的には、コードブック 1 中でエッジ部分用に用意されている 0 番から 6 3 4 8 7 番までの代表ベクトル群 Y_{j-edge} ($j-edge: 0 \sim 6 3 4 8 7$) とのマッチングを行い、その中で最も当該画像ベクトルに近いと思われる代表ベクトル Y_{k-edge} を選び、当該画像ベクトル・データ X_{i-edge} を、選択した代表ベクトルの符号化コード k に置き換える。

【0019】この場合のマッチングとは、例えばベクトル量子化による符号化を行いたい画像ベクトルと、マッチングの対象となるコードブック内の各代表ベクトルとのノルムをそれぞれ求め、最もノルムが小さい代表ベクトルを、当該画像ベクトルに最も近い代表ベクトルとして選択することである。ノルムとは、各次元成分について当該画像ベクトルの値と代表ベクトルの値との差を求め、それらの各次元の差分値の 2 乗を足し合わせた値のことをいう。

【0020】次に、フラット部分と判定されたブロックについては、当該ブロックの各成分について、16 画素の平均値を求め、それを 3 次元の画像ベクトル X_{i-flat} としてベクトル量子化を行う。すなわち、コードブック 1 内の 6 3 4 8 8 番から 6 5 5 3 5 番までの代表ベクトル Y_{j-flat} ($j-flat: 6 3 4 8 8 \sim 6 5 5 3 5$) とのマッチングを行い、最もノルムが小さい代表ベクトル Y_{l-flat} の符号化コード l に置き換える。

【0021】従って、フラット部分の場合は、ノルムの計算がエッジ部分の 48 次元に対し 3 次元で済み、しかもマッチングの回数がエッジ部分の 6 3 4 8 8 回に対し、2048 回で済むので、エッジ部分の場合に比べて符号化時間の短縮が図れる。しかも、大きな冗長度にあわせて、情報をより少なくすることができる。

【0022】復号化の際には、符号化コードを圧縮データの先頭にあるものから順にコードブック 1 を参照して、コードブック 1 からその符号化コードの代表ベクトルデータを取り出す。コードが 0 番から 6 3 4 8 7 番までのどれかであれば、そのブロックはエッジ部分であるので、16 画素分すべての情報が取り出せる。その 16 画素分の情報を復号画像データとしてそのまま使用する。符号化コードが 6 3 4 8 8 番から 6 5 5 3 5 番までのどれかであれば、そのブロックはフラット部分であるので、1 画素分の情報のみが取り出せる。この場合には、例えば復元された 1 画素分のデータをランダム関数である設定された値の範囲内で振りながら、当該ブロック内の全 16 画素に当てはめる。

【0023】この様に、エッジ部分と判定されたブロックについては、48 次元のベクトル量子化を行い、フラット部分と判定されたブロックについては 3 次元のベクトル量子化という様に、エッジ部分とフラット部分と

で、それぞれの画像の性質に合った適正な冗長削減度を持つベクトル量子化の符号化方法を選択することができ、圧縮画像の画質を大きく落とすことなく全体の冗長度を削減できる。また、上述したように、フラット部分のノルム計算がエッジ部分に比べて少なくできるため、符号化時間の短縮が図れる。なお、本実施の形態の場合の符号化圧縮率は、 4×4 画素ブロック 1 つについて、 $2 \text{ バイト符号化コード} \div 48 \text{ バイト原画像データ} = 1/24$ となる。

【0024】次に、本発明の第 2 の実施の形態について、図 2 を用いて説明する。ここで、図 2 は、本発明の第 2 の実施の形態における画像符号化方法のプロセス説明図である。図 3 に示すように、コードブック 4 には、 4×4 画素エッジ部分の代表ベクトル群・ 4×4 画素フラット部分の代表ベクトル群・ 8×8 画素フラット部分の代表ベクトル群をあらかじめ登録しておく。

【0025】それぞれの代表ベクトル群の数および符号は、例えば 4×4 画素エッジ部分：6 1 4 4 0 個 (0 番～6 1 4 3 9 番)・ 4×4 画素フラット部分：2 0 4 8 個 (6 1 4 4 0 番～6 3 4 8 7 番)・ 8×8 画素フラット部分：2 0 4 8 個 (6 3 4 8 8 番～6 5 5 3 5 番) とする。このようにすれば、符号を見るだけで、それが 4×4 画素のエッジ部分を表わすものか、 4×4 画素フラット部分を表わすものか、 8×8 画素フラット部分を表わすものかがわかる。

【0026】本実施の形態の場合、符号化の際には、まず、画像データを 8×8 画素のブロック X_h ($h: 1 \dots r$) 毎に順に見ていき、それが 8×8 画素でフラット部分であるかどうかを 8×8 画素フラット部分判別手段 6 により判別する。 8×8 画素フラット部分判別手段 6 は、例えば、ブロック内の全画素の輝度成分に注目し、その中から輝度の最大値と最小値を取りだしてその差分値を求め、その差分値をあらかじめ設定された閾値と比較し、差分値が閾値よりも小さければ、その 8×8 画素ブロックはフラット部分と判定し、そうでなければエッジ部分と判定する。そして、フラット部分と判定されると、コードブック 4 の 6 3 4 8 8 番から 6 5 5 3 5 番までの代表ベクトル $Y_{j-8 \times 8}$ ($j-8 \times 8: 6 3 4 8 8 \sim 6 5 5 3 5$) とのマッチングを行い、最も近い代表ベクトル $Y_{w-8 \times 8}$ を選択して、 8×8 画素の 192 バイト原画像ベクトルデータ X_h をその代表ベクトル $Y_{w-8 \times 8}$ の符号化コード w に置き換える。マッチングについては、第 1 の実施の形態において述べたものと同じである。

【0027】 8×8 画素でフラット部分と判定されなかった場合、その 8×8 画素のブロックは、縦横にそれぞれ 2 等分された 4 つの 4×4 画素ブロックに分割され、例えば左上の 4×4 画素ブロック X_{h1} 、右上の 4×4 画素ブロック X_{h2} 、左下の 4×4 画素ブロック X_{h3} 、右下の 4×4 画素ブロック X_{h4} の順に、それぞれ

符号化される。この場合には、第 1 の実施の形態において述べたのと同じ手法で符号化される。すなわちそれぞれのブロックで、まずエッジ部分なのかフラット部分なのかを 4×4 画素エッジ部分／フラット部分判別手段 3 により判別し、 4×4 画素エッジ部分と判定されたブロックについては、当該ブロックの全画素全成分 (4×4 画素 \times 3 成分) を 1 つの 48 次元の画像ベクトルとして、コードブックの 0 番から 61439 番までの代表ベクトルとのマッチングによりベクトル量子化を行い、 4×4 画素フラット部分と判定されたブロックについては、当該ブロックの各成分について 16 画素の平均値を求め、3 次元の画像ベクトルとしてコードブックの 61440 番から 63487 番までの代表ベクトルとのマッチングによりベクトル量子化を行う。

【0028】すなわち、第 2 の実施の形態の符号化においては、 8×8 画素ブロック (8×8 画素 \times 3 成分 \times 1 バイト／成分 = 192 バイト) 毎に、1 つの 2 バイト符号化コードまたは 4 つの 2 バイト符号化コードのいずれかが与えられる。復号化の際には、符号化コードを圧縮データの先頭にあるものから順にコードブックに参照していき、コードブックからその符号化コードの代表ベクトルデータを取り出す。符号化コードが 63488 番から 65535 番までのどれかであれば、そのブロックは 8×8 画素のフラット部分であるので、1 画素分の情報のみが取り出せる。この場合には、例えば復元された 1 画素分のデータをランダム関数である設定された値の範囲内で振りながら、当該ブロック内の全 64 画素に当てはめる。

【0029】符号化コードが 0 番から 63487 番までのどれかであれば、それは 4×4 画素ブロックの符号化コードであるので、続く 3 つの符号化コードも合わせて復号化したのち、 8×8 画素のブロックにそれら 4 つの復号データを展開しなければならない。この場合、先頭の 2 バイトコードは 8×8 画素ブロックを縦横にそれぞれ 2 等分した内の左上、2 番目のコードは右上、3 番目のコードは左下、4 番目のコードは右下の 4×4 画素ブロックの符号化データである。それぞれの 4×4 画素ブロックの復号化については、第 1 の実施の形態と同じである。

【0030】すなわち、0 番から 61439 番までのどれかであれば、そのブロックは 4×4 画素のエッジ部分であるので、復号された 16 画素分の情報を復号画像データとしてそのまま使用する。61440 番から 63487 番までのどれかであれば、そのブロックは 4×4 画素のフラット部分であるので、復元された 1 画素分のデータをランダム関数である設定された値の範囲内で振りながら、当該ブロック内の全 16 画素に当てはめる。

【0031】この様に、第 2 の実施の形態においては、圧縮画像の画質を大きく落とすことなく画像全体の冗長度を削減できるということと、符号化時間の短縮が図れ

るということの、2 つの第 1 の実施の形態で得られる効果に加えて、圧縮率を上げることができるという第 3 の効果が得られる。第 2 の実施の形態の符号化圧縮率は、 8×8 画素ブロックを 1 つの 2 バイトコードで符号化できた場合には、2 バイト符号化コード \div 192 バイト原画像データ = $1/96$ となり、4 つの 2 バイトコードで符号化された場合には、2 バイト符号化コード \times 4 \div 192 バイト原画像データ = $1/24$ となる。

【0032】従って、画像全体としては、最高 $1/192$ から最低 $1/24$ までのいずれかの符号化圧縮率が得られる。画像全体が全て 8×8 画素フラット部分と判定された時に最高圧縮率 $1/192$ が得られ、全て 4×4 画素エッジ部分またはフラット部分と判定されたときに最低圧縮率 $1/24$ が得られる。圧縮率は、 8×8 画素フラット部分と判定されるブロックが全体のどれ位を占めるかにより決まる。

【0033】以上のように、本実施の形態では、それぞれの画像ベクトル毎にエッジ部分なのかフラット部分なのかを判別するエッジ部分／フラット部分判別手段 3、6 を有し、エッジ部分・フラット部分それぞれに適したブロック内の画素数および次元数をあらかじめ定めておき、エッジ部分／フラット部分判別手段 3、6 の判定結果に従って、ブロック内の符号化の画素数および次元数を選択し、ベクトル量子化を行うようにすることにより、圧縮画像の画質を大きく落とすことなく画像全体の冗長度を削減できる。また、フラット部分の符号化の際にブロック内画素全体の平均値画素を用いることにより、ベクトル量子化の符号化時間を従来よりも短縮することができる。さらに、複数のブロックのフラット部分を合わせたものがフラット部分であると判定された場合に、それらのブロック全体についての平均値画素を用いることにより、画像データの符号化圧縮率をさらに向上できる。

【0034】なお、本実施の形態ではコードブックが 1 つである場合を取り上げたが、本発明には、2 つ以上のコードブックを用いる場合も包含される。例えば、 4×4 画素エッジ部分・ 4×4 画素フラット部分・ 8×8 画素フラット部分についてそれぞれ別にコードブックを作成しても差し支えない。

【0035】

【発明の効果】本発明によれば、圧縮画像の画質を大きく落とすことなく画像全体の冗長度を削減できる。また、ベクトル量子化の符号化時間を従来よりも短縮することができる。さらに、画像データの符号化圧縮率を向上させることができる。

【図面の簡単な説明】

【図 1】本発明の第 1 の実施の形態における画像符号化方法のプロセス説明図

【図 2】本発明の第 2 の実施の形態における画像符号化方法のプロセス説明図

【図 3】従来の画像符号化方法のプロセス説明図

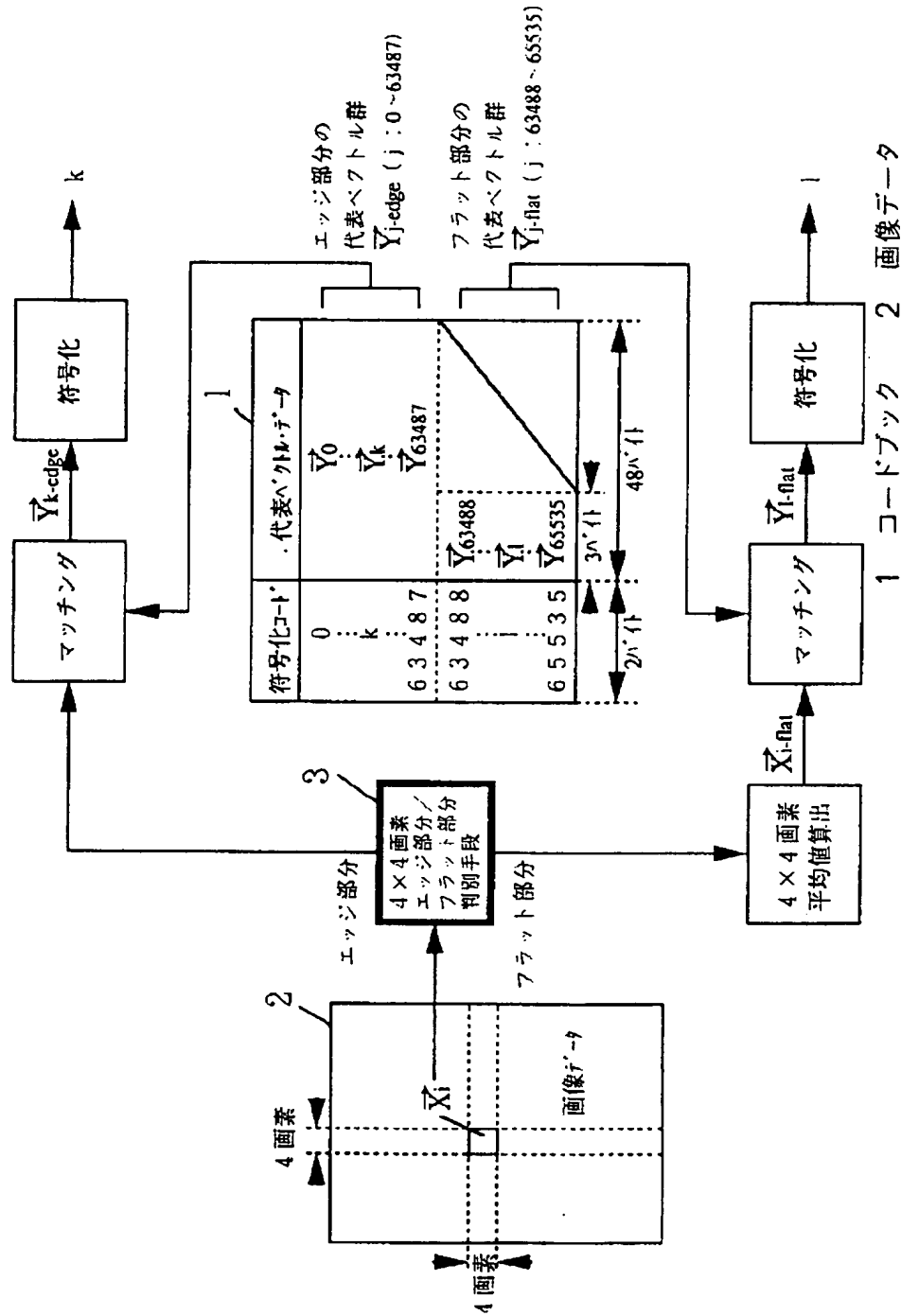
【符号の説明】

1 コードブック

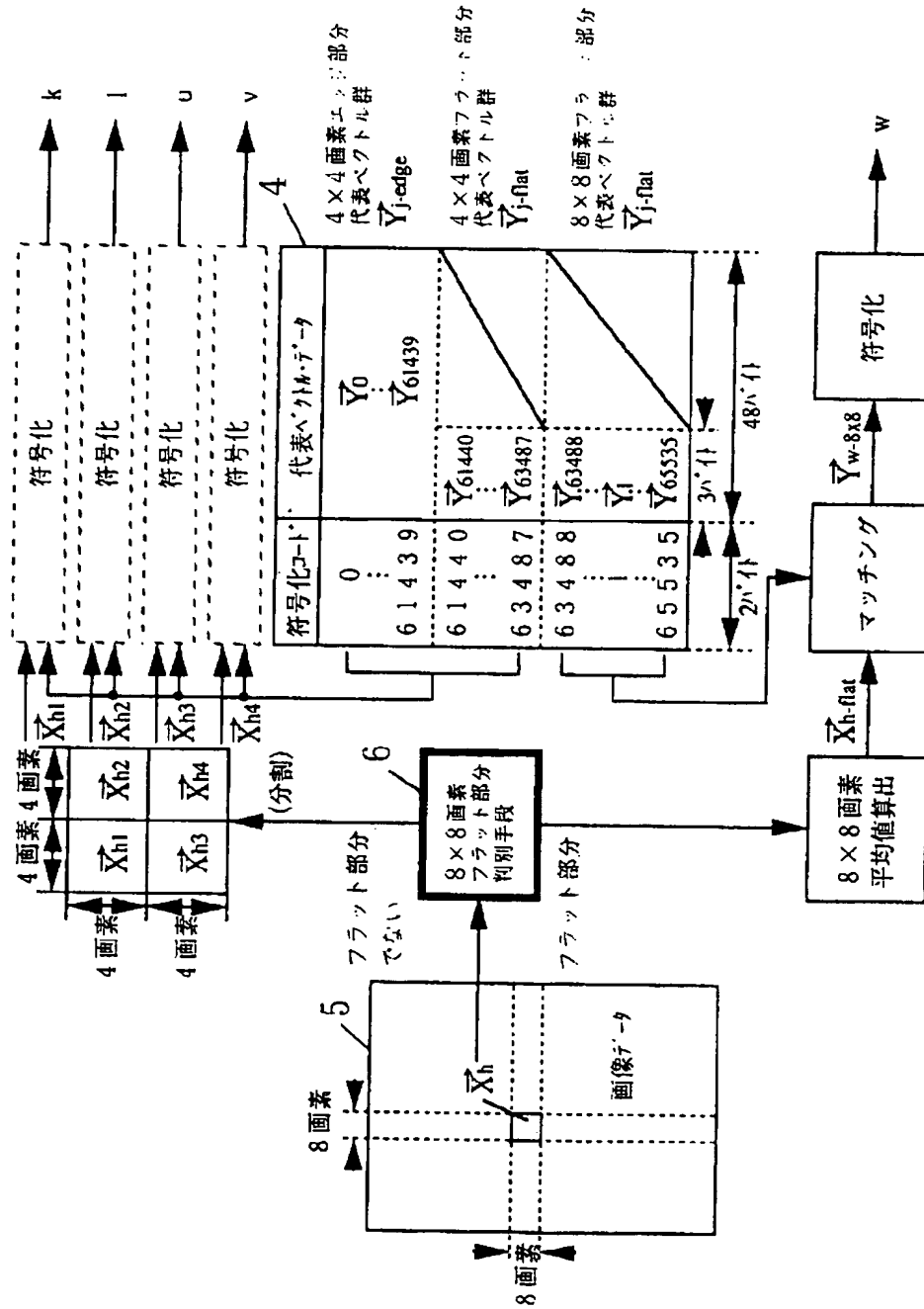
2 画像データ

3 4×4画素エッジ部分／フラット部分判別手段

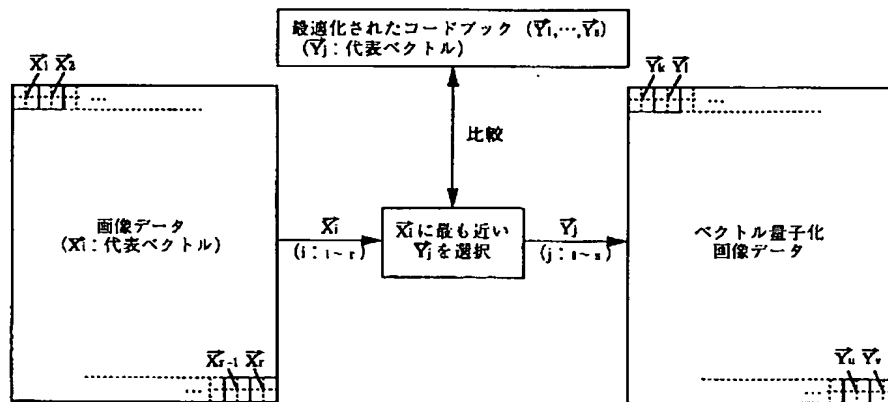
【図 1】



4×4画素フランク部分
代表ベクトル群
 $\vec{Y}_j: \text{flat}$



【図 3】



フロントページの続き

(72)発明者 小林 正明
 大阪府門真市大字門真1006番地 松下電器
 産業株式会社内